Advanced Regression: 4b Machine learning: Ensemble methods

Garyfallos Konstantinoudis

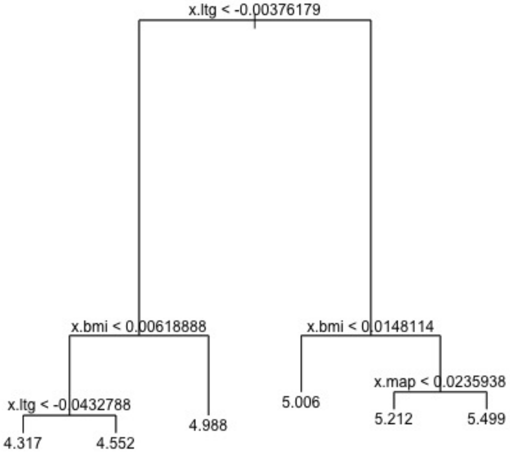
Epidemiology and Biostatistics, Imperial College London

14th March 2023

Decision trees and ensemble methods

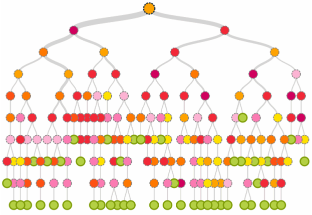
- **Decision tree**: A single tree
- **Bagging**: A meta-algorithm over trees
- **Random forest**: A meta-algorithm over random trees
- **Boosting**: A meta-algorithm over sequential trees

Decision trees: An introduction



Decision trees: An introduction

- Decision trees are drawn upside down.



Decision trees: An introduction

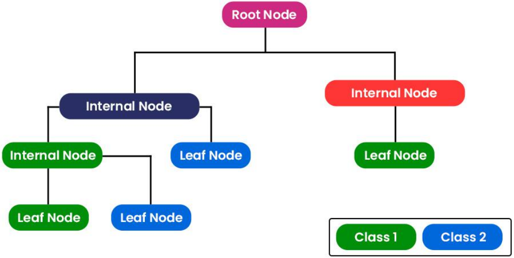
Notation:

- **Nodes or splits**: Points along the tree where the predictor space is split.
- **Leaves**: Terminal nodes
- **Branch**: Segments of a tree that connect the nodes

Outcomes:

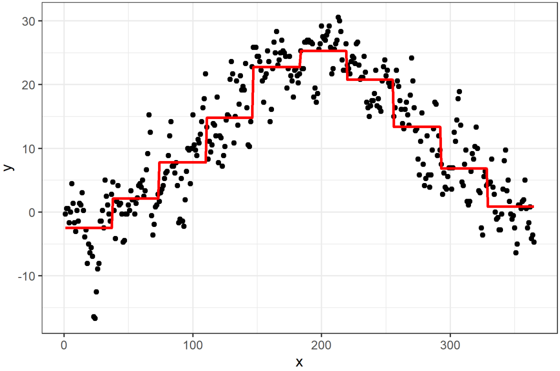
- **Quantitative**: Regression trees
- **Categorical**: Classification trees
  Considering $k = K$ categories

Decision trees: An introduction



Decision trees: Another example

Idea: Create dummies for deciles of x_i



Problem: How to select the partition?

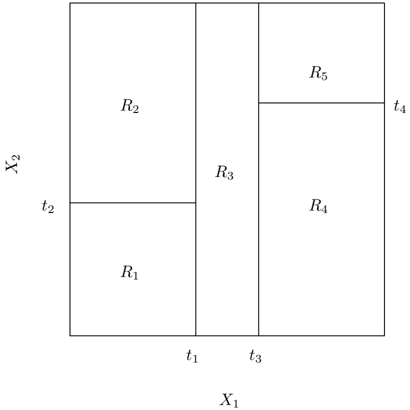
How to fit a decision tree?

1. Divide the predictor space $(x_1, x_2, ..., x_p)$ into J distinct and non-overlapping regions, $r_1, r_m, ..., r_M$, where $m \in 1, ..., M$.
2. For every observation that falls in the same region r_m we make the same prediction based on the mean (median) of all observations in region r_m.
3. Define regions $r_1, r_2, ..., r_M$ to minimise the residual sum of squares

$$RSS = \sum_{m=1}^{M} \sum_{i \in m} (y_i - \bar{y}_m)^2$$

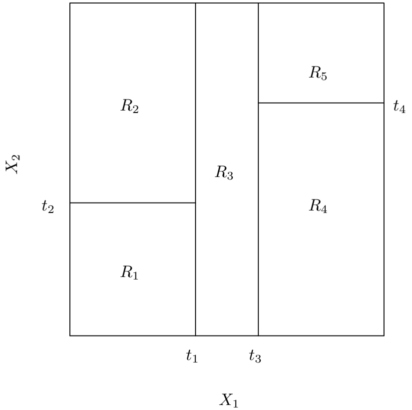
- Algorithm: Recursive binary splitting

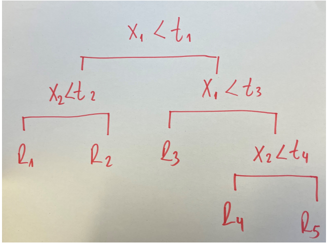
Exercise: Reconstruct the tree



- Assume we have two variables, x_1 on the x-axis and x_2 on the y-axis.
- r_1 to r_5 map out a partition.
- t_1 to t_4 are the split values.
- Reconstruct the respective tree.

Exercise: Reconstruct the tree



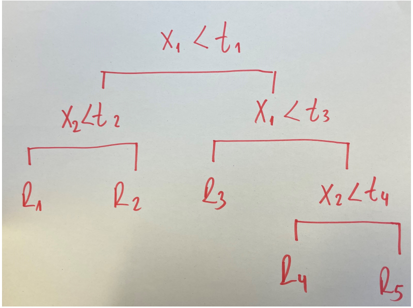


Implementation

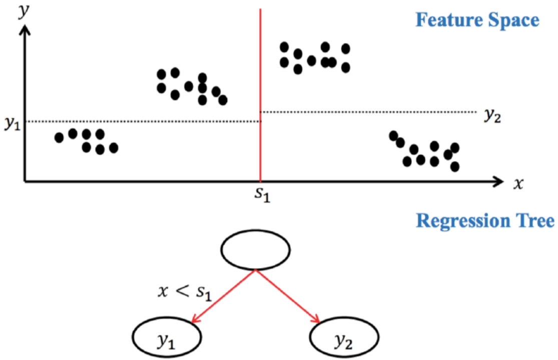
- For each variable x_k
  - Find the optimal cutoff point t:

$$\min_s \mathrm{MSE}(y_i | x_{ik} < t) + \mathrm{MSE}(y_i | x_{ik} \geq t)$$

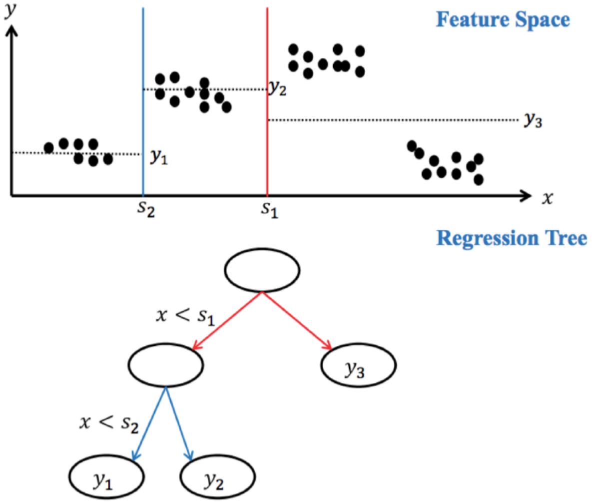
- Choose variable yielding lowest MSE
- Stop when MSE gain is too small



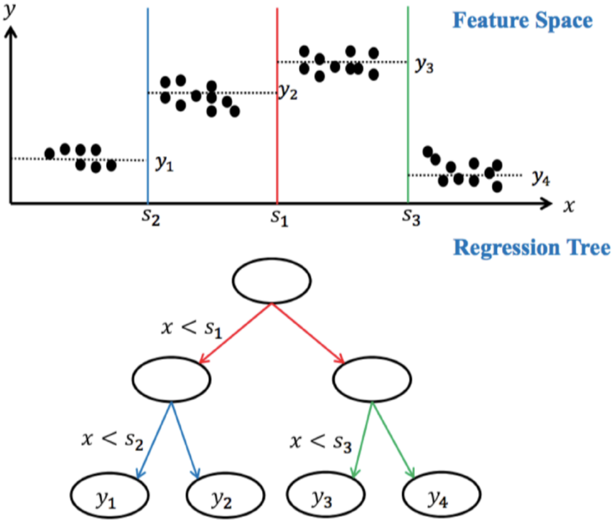
Example 1



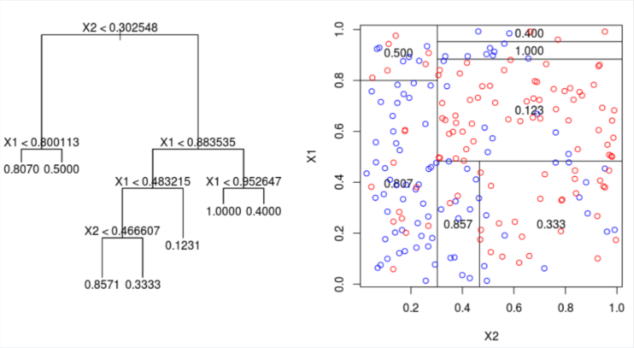
Example 1


Feature Space
y
x
y_1
y_2
y_3
s_2
s_1
Regression Tree
$x < s_1$
$x < s_2$
y_1
y_2
y_3

Example 1


y
Feature Space
y3
y2
y1
y4
x
s2
s1
s3
Regression Tree
x < s1
x < s2
x < s3
y1
y2
y3
y4

Example 2


X2 < 0.302548
X1 < 0.800113
0.8070 0.5000
X1 < 0.883535
X1 < 0.483215
X1 < 0.952647
1.0000 0.4000
X2 < 0.466607
0.1231
0.8571 0.3333
0.400
1.000
0.500
0.123
0.807
0.857
0.333
X1
X2
0.0
0.2
0.4
0.6
0.8
1.0

Measures for model fit (node impurity)

- Classification trees:
  - **Gini index** of leaf m:

$$G_m = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}),$$

    p_{mk}: proportion of observations in region R_m of class k
  - **Entropy** of leaf m

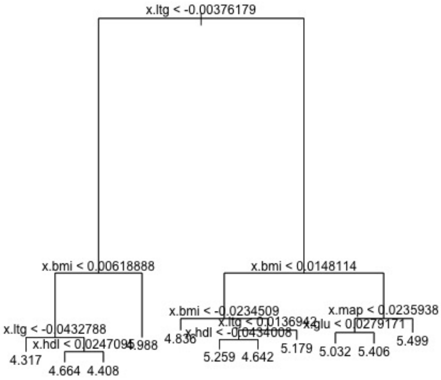
$$D_m = -\sum_{k=1}^{K} p_{mk} \log(p_{mk})$$

- Regression trees: **Deviance** in leaf m

$$dev_m = \sum_{i \in m} (y_i - \mu_m)^2$$

  where
  - $i \in m$: Individuals in leaf m
  - μ_m: Mean in leaf m

Overfitting



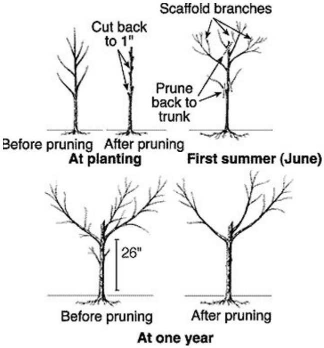
- Regression trees tend to overfit.
- In principle they could assign each observation to one leaf.

Tree pruning

- A smaller tree with fewer splits may generalise better to new observations.
- Solution: Pruning
- Cost complexity pruning or weakest link pruning: Find tree T

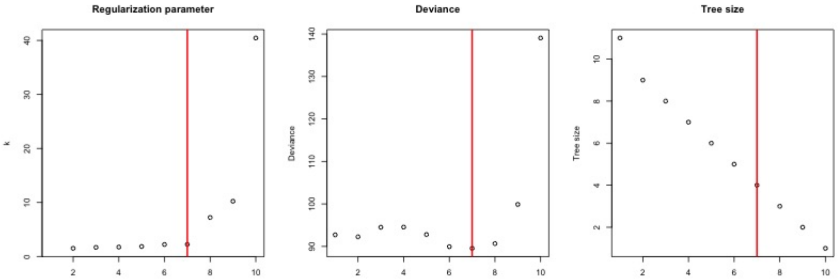
$$\underset{T}{argmin} \sum_{m=1}^{|T|} \sum_{i \in m} (y_i - \mu_m)^2 + \alpha \mid T \mid$$

where $\mid T \mid$ is the number of leaves and α a regularisation parameter.



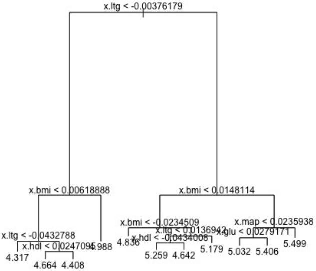
Tree pruning

- Select the regularisation parameter α that produces the tree with the lowest node impurity (measured by deviance below) as evaluated by cross-validation.



tree function in library(tree)

- Tree fit: `tree.out = tree(y ~ x)`
  `plot(tree.out)`
  `text(tree.out)`
- Parameters for tree: `tree.control(nobs, mincut = 5, minsize = 10, mindev = 0.01)`



tree function in library(tree)

- Cross-validation:

  `cv.tree(tree.out)`

```
> cv.out
$size
 [1] 11  9  8  7  6  5  4  3  2  1

$dev
 [1]  89.80061  87.15320  88.05572  88.55791  88.91603  85.55022  85.55193
 [8]  88.70582  99.30189 138.24029

$k
 [1]      -Inf  1.514803  1.688946  1.745441  1.849966  2.217092  2.240246
 [8]  7.223565 10.232546 40.443454

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```

- Pruning by keeping only best leaves:

  `prune.tree(tree.out, best)`

- Pruning by cost parameter k:

  `prune.tree(tree.out, k)`

Overview decision trees

Advantage:

- Interpretability
- Intuitive, mirror human decision making
- Allowing for non-linear effects

Disadvantages:

- Overfitting is an issue
- Highly instable and variable, small changes in the input data can cause big changes in the tree structure
- Minimal bias, but high variance

Ensemble methods

- Fit not one, but multiple trees.